

**(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION
EN MATIÈRE DE BREVETS (PCT)**

**(19) Organisation Mondiale de la Propriété
Intellectuelle**
Bureau international



(43) Date de la publication internationale
13 octobre 2005 (13.10.2005)

PCT

(10) Numéro de publication internationale
WO 2005/096135 A2

(51) Classification internationale des brevets⁷ : **G06F 7/52**

(72) Inventeurs; et

(21) Numéro de la demande internationale :

(75) Inventeurs/Déposants (pour US seulement) : **GIRAUT, Marc** [FR/FR]; 4, rue Viviane, F-14000 Caen (FR). **LEFRANC, David** [FR/FR]; Résidence Stéphanolyse, 7, rue des Tilleuls, F-14000 Caen (FR).

PCT/FR2005/000443

(22) Date de dépôt international :

24 février 2005 (24.02.2005)

(74) Mandataires : **LOISEL, Bertrand** etc.; Cabinet Plasseraud, 65/67, rue de la Victoire, F-75440 Paris Cedex 09 (FR).

(25) Langue de dépôt :

français

(81) États désignés (sauf indication contraire, pour tout titre de protection nationale disponible) : AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM,

(26) Langue de publication :

français

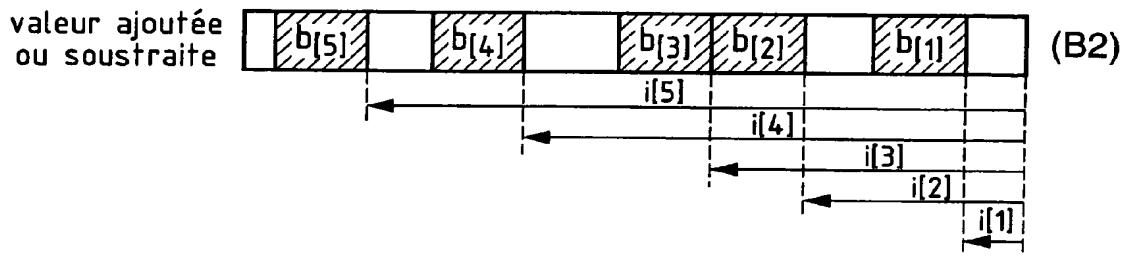
(30) Données relatives à la priorité :

0402146 2 mars 2004 (02.03.2004) FR

[Suite sur la page suivante]

(54) Title: METHOD AND DEVICE FOR PERFORMING A CRYPTOGRAPHIC OPERATION

(54) Titre : PROCEDE ET DISPOSITIF POUR ACCOMPLIR UNE OPERATION CRYPTOGRAPHIQUE



(B1) ... RANDOM

(B2) ... INCREMENT OR SUBTRACTED VALUE

WO 2005/096135 A2

(57) Abstract: The inventive method for performing a cryptographic operation by a device controlled by the security application executed outside thereof consists in producing a cryptographic value (y) in the device by a calculation comprising at least one multiplication between first and second factors containing a security key (s) associated with the device and a challenge number (c) provided by the security application. The first multiplication factor comprises a determined number of bits (L) in a binary representation. The second factor is constrained in such a way that it comprises; in a binary representation, several bits at 1 with a sequence of at least L-1 bits at 0 between each pair of consecutive bits to 1 and the multiplication is carried out by assembling the binary versions of the first factor shifted according to positions of the bits at 1 of the second factor, respectively.

[Suite sur la page suivante]



TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) États désignés (sauf indication contraire, pour tout titre de protection régionale disponible) : ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Déclaration en vertu de la règle 4.17 :

— relative à la qualité d'inventeur (règle 4.17.iv)) pour US seulement

Publiée :

— sans rapport de recherche internationale, sera republiée dès réception de ce rapport

En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

(57) Abrégé : Pour accomplir une opération cryptographique dans un dispositif sous le contrôle d'une application de sécurité exécutée en dehors du dispositif, on produit une valeur cryptographique (y) dans le dispositif, par un calcul comprenant au moins une multiplication entre des premier et second facteurs incluant une clé secrète (s) associée au dispositif et un nombre (C) dit challenge fourni par l'application de sécurité. Le premier facteur de la multiplication comprend un nombre de bits déterminé L en représentation binaire. On contraint le second facteur pour qu'il comprenne, en représentation binaire, plusieurs bits à (1) avec, entre chaque paire de bits à (1) consécutifs, une séquence d'au moins L-1 bits à (0), et en ce que la multiplication est réalisée en assemblant des versions binaires du premier facteur respectivement décalées conformément aux positions des bits à (1) du second facteur.

PROCÉDÉ ET DISPOSITIF POUR ACCOMPLIR UNE OPÉRATION CRYPTOGRAPHIQUE

La présente invention concerne un procédé pour accomplir une opération cryptographique dans un dispositif dans le cadre d'une application de sécurité. En particulier, l'invention se rapporte aux procédés cryptographiques de protection contre la fraude d'une puce électronique dans des transactions entre une application externe et la puce.

L'invention trouve une application très avantageuse en ce qu'elle permet de protéger contre la fraude des puces à circuit intégré à logique câblée, notamment les puces qui équipent les cartes prépayées utilisées dans des transactions diverses telles que l'établissement de communications téléphoniques, le paiement d'objets dans un distributeur automatique, la location d'emplacements de stationnement à partir d'un parcmètre, le paiement d'un service comme un transport public ou comme la mise à disposition d'infrastructures (péage, musée, bibliothèque,...), ou les puces qui équipent les étiquettes radiofréquences ("RFID tags") utilisées dans le traçage de palettes, de produits de grande consommation, de billets de banque, etc...

Actuellement, les puces à logique câblée sont susceptibles de subir différents types de fraude. Un premier type de fraude consiste à dupliquer sans autorisation la carte, le terme clonage étant souvent utilisé pour caractériser cette opération. Un deuxième type de fraude consiste à modifier les données attachées à une carte, en particulier le montant du crédit inscrit dans la carte. Pour lutter contre ces fraudes, il est fait appel à la cryptographie, d'une part pour assurer l'authentification de la carte au moyen d'une authentification et/ou pour assurer l'authentification des données au moyen d'une signature numérique et, d'autre part pour assurer le cas échéant la confidentialité des données au moyen d'un chiffrement. La cryptographie met en jeu deux entités, qui sont dans le cas de l'authentification un vérificateur et un objet à vérifier, et elle peut être soit symétrique, soit asymétrique. Lorsqu'elle est symétrique (ou "à clé secrète", les deux termes étant synonymes), les deux entités partagent exactement la même information, en particulier une clé secrète. Lorsqu'elle est

- 2 -

asymétrique (ou "à clé publique", les deux termes étant synonymes) une des deux entités possèdent une paire de clés dont l'une est secrète et l'autre est publique ; il n'y a pas de clé secrète partagée. Dans de nombreux systèmes, notamment lorsque la puce est de type "à logique câblée", seule la 5 cryptographie symétrique est mise en oeuvre avec des cartes prépayées, car la cryptographie asymétrique reste lente et coûteuse. Les premiers mécanismes d'authentification développés en cryptographie symétrique consistent à calculer une fois pour toutes une valeur d'authentification, différente pour chaque carte, à la stocker dans la mémoire de la carte, à la lire à chaque transaction et à la 10 vérifier en interrogeant une application du réseau supportant la transaction où les valeurs d'authentification déjà attribuées sont soit stockées soit recalculées. Ces mécanismes assurent une protection insuffisante parce que la valeur d'authentification peut être espionnée, reproduite et rejouée frauduleusement 15 étant donné qu'elle est toujours la même pour une carte donnée, permettant ainsi de réaliser un clone de cette carte. Pour lutter contre les clones, les mécanismes d'authentification passifs de cartes sont remplacés par des mécanismes d'authentification actifs qui peuvent en outre assurer l'intégrité des données.

Le principe général des mécanismes d'authentification actifs 20 symétriques est le suivant : lors d'une authentification, la puce électronique et l'application calculent une valeur d'authentification qui est le résultat d'une fonction appliquée à une liste d'arguments déterminée à chaque authentification. La liste d'arguments peut comprendre un aléa, c'est-à-dire une donnée déterminée par l'application à chaque authentification, une donnée 25 contenue dans la puce électronique et une clé secrète connue de la puce électronique et de l'application. Lorsque la valeur d'authentification calculée par la puce électronique est identique à la valeur d'authentification calculée par l'application, la puce électronique est jugée authentique et la transaction entre la puce électronique et l'application est autorisée.

30 De tels mécanismes d'authentification sont largement connus mais la plupart exigent des capacités de calcul au moins égales à celles dont dispose un microprocesseur. Ces mécanismes conviennent donc aux cartes à

microprocesseur, mais rarement aux puces à logique câblée, lesquelles disposent de moyens de calcul beaucoup plus rudimentaires.

Un premier pas a été effectué lorsque des mécanismes actifs d'authentification symétriques ont pu être intégrés dans des puces à logique câblée. Par exemple, FR-A-2 826 531 décrit un procédé permettant de spécifier de tels mécanismes. On notera que la valeur d'authentification produite par ces mécanismes peut aussi être interprétée comme une séquence de bits pseudo-aléatoires et, en faisant varier au moins l'un des paramètres d'entrée, le procédé de calcul de la valeur d'authentification devient alors un procédé de génération de bits pseudo-aléatoires.

Cependant, les mécanismes à clé secrète imposent que l'unité de vérification, en charge de l'authentification de la puce, par exemple présente dans un téléphone public, un terminal de paiement électronique, ou encore un portillon de transport en commun, connaisse la clé secrète détenue par ladite puce. Il en découle un inconvénient majeur, à savoir que si l'on souhaite que cette unité puisse authentifier n'importe quelle puce mise en relation avec l'application, soit elle doit stocker les clés secrètes de toutes les puces, soit elle doit stocker une clé de base, appelée aussi clé-mère ou clé-maître, permettant de retrouver la clé secrète de n'importe quelle puce. Dans les deux cas, chacune de ces unités stocke suffisamment d'information pour pouvoir retrouver les clés secrètes de toutes les puces émises, et stocke donc suffisamment d'information pour pouvoir fabriquer des clones de n'importe laquelle d'entre elles. Il s'ensuit qu'une intrusion réussie contre n'importe laquelle des unités de vérification anéantirait la sécurité de l'application dans son ensemble.

Il existe donc un besoin impérieux de pouvoir intégrer un mécanisme actif d'authentification à clé publique dans une puce à logique câblée, notamment dans les applications déployant un grand nombre de puces, ce qui est généralement le cas des applications utilisant des puces à logique câblées, car elles sont très bon marché.

Il n'existe pas de tels mécanismes car les schémas à clés publiques

requièrent généralement de nombreuses opérations portant sur de grands nombres, ce qui les rend a priori inadaptés à une intégration dans des puces à logique câblée, dans lesquelles la surface de silicium est extrêmement réduite, et dont la logique de calcul se réduit au câblage d'opérations extrêmement élémentaires. De surcroît, ces opérations élémentaires sont effectuées généralement en série, en ce sens que les opérandes sont introduits séquentiellement bit à bit, et que cette introduction modifie progressivement l'état d'un registre interne dont la valeur finale sert de base au calcul du résultat de la fonction.

Par exemple, il est connu de calculer une valeur d'authentification y en effectuant l'opération $y = r + s.c$ (ou $y = r - s.c$), où r désigne un aléa, s une clé secrète appartenant à une paire de clés asymétrique (s, p) , et c est une valeur dite "challenge" choisie par l'application, qui ne peut pas dépasser un certain seuil, la valeur de r devant toujours rester supérieure au nombre qui lui est ajouté (ou soustrait). Ceci implique l'exécution d'au moins une multiplication entre entiers a priori quelconques s, c . Or une telle multiplication est une opération relativement complexe, hors de portée de la plupart des puces à logique câblée.

Un but de la présente invention est de définir un procédé de calcul d'une valeur cryptographique, par exemple d'authentification, qui soit tel que la puce n'ait pas à effectuer explicitement une opération de multiplication et que le niveau de sécurité obtenu soit relativement important, par exemple de l'ordre de 32 bits, ce qui est un niveau de sécurité extrêmement répandu, notamment dans la protection de transactions financières.

L'invention propose un procédé pour accomplir une opération cryptographique dans un dispositif sous le contrôle d'une application de sécurité, dans lequel on produit une valeur cryptographique dans le dispositif, par un calcul comprenant au moins une multiplication entre deux facteurs incluant une partie au moins d'une clé secrète associée au dispositif. Selon l'invention, un premier des deux facteurs de la multiplication a un nombre de bits déterminé L en représentation binaire, et on contraint le second des deux facteurs de la multiplication pour qu'il comprenne, en représentation binaire,

plusieurs bits à 1 avec, entre chaque paire de bits à 1 consécutifs, une séquence d'au moins $L-1$ bits à 0. La multiplication n'a alors pas besoin d'être exécutée selon un algorithme complexe. Il suffit de la réaliser en assemblant des versions binaires du premier facteur respectivement décalées 5 conformément aux positions des bits à 1 du second facteur.

Selon d'autres caractéristiques intéressantes du procédé de l'invention:

- la clé secrète fait partie d'une paire de clés cryptographiques asymétriques associée au dispositif;
- le dispositif comprend une puce incluant une logique câblée pour produire la valeur cryptographique; 10
- le calcul de la valeur cryptographique comprend en outre une addition ou une soustraction entre un nombre pseudo-aléatoire et le résultat de la multiplication;
- les premier et second facteurs et le nombre pseudo-aléatoire sont dimensionnés pour que le nombre pseudo-aléatoire soit supérieur au résultat de la multiplication, le nombre de bits à 1 du second facteur pouvant notamment être choisi au plus égal au plus grand entier inférieur ou égal à s_1/L , où s_1 est un seuil prédéfini inférieur au nombre de bits du nombre pseudo-aléatoire en représentation binaire; 15
- les deux facteurs de la multiplication incluent, en plus de ladite partie de la clé secrète, un nombre fourni au dispositif par l'application de sécurité, laquelle est exécutée en dehors du dispositif;
- ladite partie de la clé secrète est soit le premier soit le second facteur de la multiplication; 20
- lorsque ladite partie de la clé secrète est le premier facteur de la multiplication, lesdites versions binaires sont par exemple disposées dans des intervalles respectifs de même taille en bits, ladite taille correspondant à la taille totale d'un espace utilisable divisée par le nombre de bits à 1 du second facteur de la multiplication, chaque version binaire étant placée dans son intervalle respectif en fonction d'un décalage défini par l'application de sécurité; 25
- lorsque ladite partie de la clé secrète est le second facteur de la multiplication, lesdites versions binaires sont par exemple disposées dans des intervalles respectifs de même taille en bits, ladite taille correspondant à la taille totale d'un espace utilisable divisée par le nombre de bits à 1 du premier facteur de la multiplication, chaque version binaire étant placée dans son intervalle respectif en fonction d'un décalage défini par l'application de sécurité; 30

- lorsque la clé secrète est le second facteur de la multiplication, elle peut être stockée dans un support de mémoire du dispositif en codant des nombres de bits séparant respectivement des bornes inférieures d'intervalles de $(S-1)/(n-1)$ bits et des bornes inférieures de blocs de bits alloués au premier facteur de la multiplication et disposés chacun dans les intervalles associés, S étant le nombre de bits de la clé secrète et n le nombre de bits à 1 de la clé secrète;
- 5 - en variante, on peut stocker cette clé secrète dans le support de mémoire du dispositif en codant des nombres de bits, chacun représentatif du nombre de bits séparant deux blocs de bits successifs alloués au premier facteur de la multiplication;
- 10 - lorsque ladite partie de la clé secrète est le second facteur de la multiplication, elle est stockée dans un support de mémoire du dispositif en codant les positions de ses bits à 1;
- 15 - lorsque ladite partie de la clé secrète est le second facteur de la multiplication, le premier facteur peut être un nombre pseudo-aléatoire généré dans le dispositif, la valeur cryptographique étant produite en tant que signature électronique;
- dans une autre application, la valeur cryptographique est produite pour 20 authentifier le dispositif dans une transaction avec l'application de sécurité exécutée en dehors du dispositif.

L'invention propose également un dispositif à fonction cryptographique, comprenant des moyens d'interface avec une application de sécurité et des moyens de calcul pour produire une valeur cryptographique, les moyens de calcul comprenant des moyens de multiplication entre deux facteurs incluant une partie au moins d'une clé secrète associée au dispositif. Selon l'invention, un premier des deux facteurs de la multiplication a un nombre de bits déterminé L en représentation binaire, et le second des deux facteurs de la multiplication est contraint pour comprendre, en représentation binaire, plusieurs bits à 1 avec, entre chaque paire de bits à 1 consécutifs, une séquence d'au moins L-1 bits à 0. Les moyens de multiplication comprennent des moyens pour assembler des versions binaires du premier facteur

respectivement décalées conformément aux positions des bits à 1 du second facteur.

5 Un avantage de la présente invention est d'obtenir un bon niveau de sécurité des dispositifs utilisant la cryptographie pour se protéger contre la fraude, particulièrement lors des transactions entre une puce électronique à logique câblée et une application de sécurité externe à la puce électronique.

10 Un autre avantage de l'invention est sa simplicité de mise en œuvre puisqu'elle ne nécessite pas de moyens de multiplication, coûteux et difficiles à mettre en œuvre dans une puce électronique du fait de sa taille réduite, les moyens de multiplication étant remplacés par des moyens d'addition pour sommer les décalés de la clé secrète ou du challenge. En effet, la multiplication d'un entier par une puissance de deux revient à décaler vers la gauche les bits 15 de la décomposition binaire.

D'autres particularités et avantages de la présente invention apparaîtront dans la description ci-après d'exemples de réalisation non limitatifs, en référence aux dessins annexés, dans lesquels :

- la figure 1 illustre un organigramme décrivant un mode de réalisation d'une authentification cryptographique ;
- la figure 2 illustre un mode de réalisation du procédé de l'invention;
- 20 - la figure 3 illustre une première variante de réalisation du procédé de l'invention ;
- la figure 4 illustre une seconde variante de réalisation du procédé de l'invention;
- la figure 5 illustre une première variante de stockage d'une clé secrète selon l'invention ;
- 25 - la figure 6 illustre une seconde variante de stockage d'une clé secrète selon l'invention ;
- la figure 7 illustre un exemple de dispositif à fonction cryptographique mettant en œuvre le premier mode de réalisation de l'invention; et

- la figure 8 illustre un exemple d'utilisation d'un dispositif à fonction cryptographique selon l'invention.

La figure 1 illustre un organigramme d'un procédé pour exécuter une opération cryptographique dans un dispositif, particulièrement une puce électronique à logique câblée, sous le contrôle d'une application de sécurité exécutée en dehors du dispositif.

Notamment, un tel procédé est appliqué dans des transactions entre la puce électronique et l'application. Mais il peut aussi bien être utilisé pour le calcul d'une valeur cryptographique constituant une signature numérique.

10 Dans une première étape 1, le procédé consiste à générer dans la puce électronique un nombre pseudo-aléatoire r dit aléa au moyen d'un générateur pseudo-aléatoire inclus dans la puce électronique. L'aléa r est propre à une transaction.

15 Dans une deuxième étape 2, le procédé consiste à transmettre de la puce électronique à l'application de sécurité un paramètre x relié à l'aléa r sur la base d'une fonction f telle que $x = f(r)$, f étant associée à une clé publique connue de la puce électronique et de l'application de sécurité. Des paramètres x peuvent être calculés d'avance et stockés dans une mémoire de la puce, chaque paramètre x étant associé respectivement à un nombre pseudo-aléatoire r .

20 A titre d'exemple, g étant un nombre connu de l'application de sécurité, la fonction f peut classiquement être de la forme : $x = f(r) = g^r$. Une autre possibilité, également bien connue de l'homme du métier, est d'utiliser dans la fonction f une fonction de hachage cryptographique h ainsi que des données D liées à l'application (par exemple un montant de transaction) : $x = f(r) = h(g^r, D)$.
25 D'autres exemples de fonction f sont également utilisables.

30 Dans une troisième étape 3, le procédé consiste à calculer dans la puce électronique une valeur cryptographique y , conformément au procédé de l'invention décrit plus loin, par des moyens de multiplication ayant pour paramètres d'entrée au moins l'aléa r propre à la transaction et une clé secrète

appartenant à une paire de clés asymétriques (s, p), par exemple avec $p = g^s$.

La valeur cryptographique constitue tout ou partie d'une valeur d'authentification V.

Dans une quatrième étape 4, le procédé consiste à transmettre la 5 valeur d'authentification V à l'application.

Dans une cinquième étape 5, le procédé consiste à vérifier par l'application de sécurité la valeur d'authentification V au moyen d'une fonction de vérification dont les paramètres d'entrée se composent de paramètres publics, contenant au moins la clé publique p. Si la valeur d'authentification est 10 validée par l'application de sécurité, les transactions sont autorisées.

La présente invention a pour but de calculer une valeur cryptographique y en effectuant une opération du type $y = r + f_1 * f_2$ (ou de façon équivalente $y = r - f_1 * f_2$) où r désigne un aléa calculé par un générateur pseudo-aléatoire inclus dans la puce électronique et (f_1, f_2) un couple de 15 facteurs incluant la clé secrète s et un nombre c dit challenge fourni par l'application de sécurité. Les représentations binaires des facteurs f_1, f_2 sont telles que le facteur f_1 a une taille en bits notée L et que les bits à 1 successifs du facteur f_2 ont toujours entre eux une séquence d'au moins $L-1$ bits à 0.

Ainsi, la multiplication des facteurs f_1, f_2 revient à sommer un nombre n 20 de versions binaires $b[1], b[2], \dots, b[n]$ du premier facteur f_1 respectivement décalées conformément aux positions des bits à 1 du second facteur f_2 . Le nombre n est dépendant du nombre de bits à 1 du second facteur f_2 . Compte

tenu de ces décalages, la somme $\sum_{h=1}^n b[h]$ est réalisée très simplement en

25 assemblant les $b[h]$ dans un nombre binaire représentant le résultat de la multiplication.

Dans la description qui suit, on décrit un premier mode de réalisation du procédé de l'invention où le facteur f_1 est la clé secrète s de L bits et le

second facteur f_2 est le challenge pseudo-aléatoire c fourni par l'application de sécurité. Ainsi la valeur cryptographique y est déterminée en effectuant l'opération:

$$y = r + s * (2^{i[1]} + \dots + 2^{i[n]})$$

5 où $i[1], \dots, i[n]$ désignent n valeurs entières choisies par l'application avec deux contraintes :

- les valeurs $i[1], \dots, i[n]$ doivent toutes rester inférieures ou égales à un seuil noté s_1 de sorte que la valeur r reste toujours supérieure au nombre qui lui est ajouté (ou soustrait), ledit seuil définissant alors un espace utilisable sur les bits de r .
- la seconde contrainte consiste à imposer que les différentes valeurs $i[1], \dots, i[n]$ soient distantes les unes des autres par au moins la taille en bits de la clé secrète.

Ainsi, la clé secrète s étant composée de L bits, et en supposant les 15 différents entiers $i[1], \dots, i[n]$ rangés par ordre croissant, l'application de sécurité fournit ces valeurs $i[1], \dots, i[n]$ telles que $i[1]+L \leq i[2], i[2]+L \leq i[3], \dots, i[n-1]+L \leq i[n]$ et $i[n] < s_1$.

Ainsi, le nombre n de bits à 1 du second facteur est choisi au plus égal au plus grand entier inférieur ou égal à s_1/L .

20 Les deux contraintes imposent à la valeur de n un maximum noté u , qui ne sera généralement pas très élevé, par exemple $u = 5$ ou 6 . Préférentiellement, la valeur de n sera prise égale au maximum u .

La figure 2 illustre un exemple d'utilisation du procédé de l'invention avec $n = 5$. Le diagramme B1 représente l'aléa r avec l'espace utilisable EU de 25 taille $L+s_1$ bits. Le diagramme B2 représente le résultat de la multiplication $s*c = s * (2^{i[1]} + \dots + 2^{i[n]})$, ajouté (ou soustrait) à r . La clé secrète s est décalée conformément aux positions des bits à 1 du challenge c . On obtient ainsi les versions binaires $b1$ à $b5$ respectivement décalées de $i[1], i[2], i[3], i[4]$ et $i[5]$ bits.

Un avantage de la présente invention repose sur le gain de sécurité obtenu avec un tel procédé, car deviner les différentes valeurs des entiers $i[h]$, et éventuellement le nombre n lui-même s'il n'est pas fixé par l'application de sécurité, est très difficile.

5 Un autre avantage est que le gain de sécurité se fait sans pour autant complexifier significativement le calcul de la valeur cryptographique y puisque la multiplication entre la clé secrète et le challenge revient à assembler des décalés (ou versions binaires) conformément à l'invention, le calcul étant effectué en série.

10 Selon une première variante du procédé, on impose que les n décalés de la clé secrète soient tous placés dans des intervalles de même taille. Cette taille est souhaitée maximale, ce qui signifie qu'elle est égale à la taille totale de l'espace utilisable EU divisée par n ou, si cette valeur n'est pas un entier, le plus grand entier qui lui soit inférieur. On notera $j[1], j[2], \dots, j[n]$ les écarts, 15 choisis par l'application, des blocs de bits correspondant aux décalés de la clé secrète avec la borne inférieure de l'intervalle dans lequel ces blocs associés sont situés.

Plus formellement, soit k la taille totale de l'espace utilisable (égale à s_1+L), n est alors le nombre de fois où l'on veut faire apparaître la clé secrète 20 de façon disjointe parmi les k bits utilisables. On suppose que k est divisible par n (dans le cas contraire, on augmente légèrement la valeur de k pour qu'il en soit ainsi).

L'espace total utilisable peut se décomposer en n intervalles de même taille contenant chacun k/n positions de la façon suivante :

25 $[0, k/n-1] \cup [k/n, 2k/n-1] \cup \dots \cup [(n-1)k/n, k-1]$

Avec l'utilisation de la présente variante, on peut démontrer qu'il y a au total $((k/n)-L+1)^n$ valeurs différentes que l'on peut ajouter à l'aléa r .

En effet, chaque intervalle contient k/n positions dont L utilisées pour écrire le décalé de la clé secrète. Il reste donc $k/n-L$ positions libres dans

chaque intervalle à repartir autour du bloc de bits utilisés par le décalé. Ainsi, pour tout h tel que $1 \leq h \leq n$, $j[h]$ est une valeur comprise entre 0 et $(k/n) - L$ et correspond au nombre de bits entre la $(h-1)(k/n)$ -ième position et la position du premier bit du décalé. Il y a donc $(k/n) - L + 1$ valeurs possibles pour chaque $j[h]$.

5 Finalement, le nombre de n -uplets $(j[1], j[2], \dots, j[n])$ différents est $((k/n) - L + 1)^n$.

En d'autres termes, la valeur cryptographique y est calculée en effectuant l'opération :

$$y = r + (2^{j[1]}.s + 2^{k/n+j[2]}.s + 2^{2k/n+j[3]}.s + \dots + 2^{(n-1)k/n+j[n]}.s)$$

où les n valeurs $j[1], j[2], \dots, j[n]$ sont choisies par l'application de sécurité de

10 façon telle que $j[h] \in [0, (k/n) - L]$ pour tout h .

Chaque version binaire $b[h]$ est donc décalée de $j[h]$ bits par rapport à la borne inférieure de l'intervalle.

Sur la figure 3, on a noté $I[h] = [(h-1)k/n, hk/n - 1]$ le h -ième intervalle dans un cas particulier de la première variante de réalisation où $n = 5$. Le

15 diagramme B3 représente l'aléa r avec l'espace utilisable EU, tandis que le diagramme B4 représente la valeur ajoutée (ou soustraite). On obtient ainsi des versions binaires $b[h]$ de la clé secrète s , à savoir $b[1], b[2], b[3], b[4]$ et $b[5]$, respectivement disposées dans les intervalles $I[h]$. Chaque version binaire $b[h]$ est décalée de $j[h]$ bits par rapport à la borne inférieure de l'intervalle $I[h]$ associé.

Selon une seconde variante du procédé de l'invention, les n valeurs d'écart $j[1], j[2], \dots, j[n]$ sont choisies par l'application de sécurité de façon à

représenter les écarts entre deux blocs consécutifs correspondant à deux décalés de la clé secrète. Comme dans la première variante, les n valeurs

25 peuvent être prises dans l'intervalle $[0, (k/n) - L]$.

La valeur cryptographique est ainsi calculée en effectuant l'opération :

$$y = r + (2^{j[1]}.s + 2^{L+j[1]+j[2]}.s + 2^{2L+j[1]+j[2]+j[3]}.s + \dots + 2^{(n-1)L+j[1]+j[2]+\dots+j[n]}.s)$$

La figure 4 illustre un exemple d'utilisation de la seconde variante du

procédé de l'invention avec $n=5$. Le diagramme B5 représente l'aléa r avec l'espace utilisable EU , tandis que le diagramme B6 représente la valeur ajoutée (ou soustraite). On obtient ainsi cinq versions binaires $b[1]$, $b[2]$, $b[3]$, $b[4]$ et $b[5]$, de la clé secrète s , décalées entre elles de $j[1]$, $j[2]$, $j[3]$, $j[4]$ et $j[5]$ bits,
5 respectivement.

Dans certains cas, on pourra relâcher quelque peu la contrainte sur le nombre de bits à 0 devant séparer deux bits à 1 consécutifs du challenge c , au prix d'une augmentation modérée de la complexité de la logique câblée de la puce, sans sortir du cadre de la présente invention. Considérons par exemple
10 le cas où la clé secrète $s = f_1$ du dispositif est un nombre de L bits avec $n = 6$.

Si on impose que deux bits à 1 consécutifs du challenge $c = 2^{i[1]} + 2^{i[2]} + 2^{i[3]} + 2^{i[4]} + 2^{i[5]} + 2^{i[6]}$ soient toujours séparés par une séquence d'au moins $L/2$ bits à 0 (c'est-à-dire que $i[1]+L/2 < i[2]$, ..., $i[5]+L < i[6] < s_1$, il est aisé de décomposer ce challenge en deux termes:
15 $c = f_2 + f'_2$, où $f_2 = 2^{i[1]} + 2^{i[3]} + 2^{i[5]}$ et $f'_2 = 2^{i[2]} + 2^{i[4]} + 2^{i[6]}$. Chacun de ces deux termes f_2 , f'_2 vérifie la condition requise pour être le second facteur de la multiplication dans le procédé selon l'invention: $i[1]+L < i[3] < i[5]-L$ pour f_2 et $i[2]+L < i[4] < i[6]-L$ pour f'_2 . L'opération cryptographique à effectuer s'écrit
20 $y = r + s.f_2 + s.f'_2$, chacune des deux multiplications $s.f_2$, $s.f'_2$ pouvant être réalisée par assemblage de versions décalées de s conformément à l'invention. La logique câblée doit alors comporter deux circuits additionneurs (ou soustracteurs).

Dans un autre mode de réalisation de l'invention, les rôles de s et de c dans la multiplication sont inversés: le premier facteur f_1 est le nombre pseudo-
25 aléatoire c , tandis que le second facteur f_2 est la clé secrète s . La valeur cryptographique y est donc calculée en effectuant l'opération:

$$y = r + c*(2^{i[1]} + \dots + 2^{i[n]})$$

où r désigne un aléa, c le challenge choisi par l'application de sécurité, et les n valeurs $i[1], \dots, i[n]$ correspondent aux positions des bits non nuls de la clé

secrète s associée au dispositif, ou d'une partie de cette clé secrète. En d'autres termes, la clé secrète s est $2^{i[1]} + 2^{i[2]} + \dots + 2^{i[n]}$. Les n valeurs $i[1], \dots, i[n]$ doivent vérifier les mêmes contraintes que celles présentées lors du premier mode de réalisation.

5 Le procédé pour le calcul d'une telle valeur cryptographique est identique à celui précédemment présenté dans le premier mode de réalisation.

Toutefois, la clé secrète $s = 2^{i[1]} + 2^{i[2]} + \dots + 2^{i[n]}$ reste identique pour toutes les authentifications, comme l'est une clé secrète classique. Deux cas peuvent se présenter pour l'utilisation de ce second mode de réalisation :

10 - la clé secrète est dédiée à une application particulière nécessitant un niveau de sécurité L_0 constant. Dans ce cas, la construction de la clé secrète s se fait en utilisant une clé secrète s avec des bits non nuls séparés par au moins (L_0-1) bits; ou

15 - la clé secrète est utilisée dans différentes applications nécessitant des niveaux de sécurité différents. Il faut alors considérer le niveau de sécurité le plus grand. En effet, dans le cas contraire, la construction d'une clé avec des bits non nuls séparés par (L_0-1) bits peut donner lieu à des challenges de taille $L_1 > L_0$, de sorte que les décalés du challenge ne seraient plus disjoints. D'où la nécessité d'avoir des bits non nuls de la clé secrète séparés par au moins $L-1$ bits, où L représente le niveau maximal de sécurité pouvant être rencontré durant l'utilisation de la clé secrète s .

25 Pour mémoriser la clé secrète, une première solution consiste à la stocker dans sa totalité. Mais étant donné la taille de la clé secrète due au niveau de sécurité élevé, des contraintes physiques, particulièrement la taille de la mémoire de la puce électronique, limitent la taille de stockage possible, notamment pour des raisons économiques.

30 Une seconde solution consiste à tirer parti de la structure de la clé secrète du type $2^{i[1]} + 2^{i[2]} + \dots + 2^{i[n]}$, en ne stockant que les positions des bits non nuls de la clé secrète. Cette seconde solution peut être améliorée, afin de

stocker encore moins de bits. Deux méthodes sont possibles.

Soit S la taille de la clé secrète ayant n bits non nuls en faisant l'hypothèse que $S-1$ soit divisible par $n-1$. Un premier bit non nul est placé à la position $S-1$ pour obtenir un secret de la taille souhaitée. Ensuite, on découpe 5 les $S-1$ autres bits de la clé secrète en $n-1$ intervalles $I[1], I[2], \dots, I[n-1]$ de $(S-1)/(n-1)$ bits.

Dans chacun des intervalles $I[h]$, on place un bloc de bits alloué au bloc de bits représentant le challenge de taille L . Chaque intervalle contient donc $(S-1)/(n-1) - L$ bits non utilisés qui se répartissent à gauche et à droite 10 du bloc de bits alloué. Au maximum, $(S-1)(n-1) - L$ bits peuvent être placés entre la borne inférieure de l'intervalle et le bloc alloué.

Ainsi, au lieu de stocker la position $i[h]$ des bits non nuls, on peut stocker le nombre de bits $p[h]$ se trouvant entre la borne inférieure de chaque intervalle et ledit bloc de bits alloué se trouvant dans l'intervalle considéré.

15 La figure 5 illustre un exemple de stockage d'une clé secrète ayant cinq bits non nuls. On stocke ainsi les nombres de bits $p[1], p[2], p[3], p[4]$ (dans l'exemple nul) et $p[5]$.

La seconde méthode consiste à stocker le nombre de bits se trouvant entre deux blocs consécutifs de bits alloués au bloc relatif au challenge et le 20 nombre de bits se trouvant à chaque extrémité (avant le premier bloc alloué et après le dernier bloc alloué).

La figure 6 illustre la présente méthode dans le cas où la clé secrète contient cinq bits non nuls. Ainsi les valeurs $p[1], p[2], p[3], p[4], p[5]$ et $p[6]$ sont stockées sur un support de mémoire de la puce électronique. Cet exemple 25 n'est pas limitatif et des valeurs $p[h]$ peuvent être nulles.

Dans le second mode de réalisation de l'invention, il est aussi possible dans certains cas de relâcher quelque peu la contrainte sur le nombre de bits à 0 devant séparer deux bits à 1 consécutifs de la clé s , au prix d'une augmentation modérée de la complexité de la logique câblée de la puce, sans 30 sortir du cadre de la présente invention.

Le procédé de l'invention est utilisable pour calculer une valeur cryptographique pour protéger contre la fraude un dispositif, particulièrement une puce électronique à logique câblée, sous le contrôle d'une application de sécurité externe au dispositif, dans les transactions entre ces deux entités.

5 Un tel procédé est également utilisable pour calculer une valeur cryptographique comme constituant d'une signature numérique. Dans ce cas, le nombre c dit challenge n'est pas fourni par l'application de sécurité mais est calculé par la puce électronique d'après un message à signer.

10 La figure 7 illustre un exemple de dispositif à fonction cryptographique mettant en œuvre le premier mode de réalisation de l'invention.

Le dispositif 10, tel qu'une puce électronique, comprend :

- un générateur de nombre pseudo-aléatoire 12 produisant un aléa r propre à une transaction, l'aléa r étant lié à un paramètre x associé ;
- une première mémoire 16 pour stocker la clé secrète s ;
- une seconde mémoire 14 pour stocker les paramètres x ;
- une interface 24 pour échanger des données avec l'application de sécurité externe, de façon connue en soi ;
- un circuit 22 de multiplication entre la clé secrète s et le challenge c ; et
- un additionneur 26 (ou un soustracteur) pour combiner arithmétiquement le résultat de la multiplication et le nombre pseudo-aléatoire issu du générateur 12.

20 Dans la réalisation illustrée par la figure 7, le circuit de multiplication 22 décale le premier facteur f_1 , particulièrement la clé secrète s , conformément aux exigences de l'invention, puis transmet séquentiellement bit par bit le résultat obtenu à l'additionneur 26.

Parallèlement, le générateur pseudo-aléatoire 12 transmet séquentiellement bit par bit l'aléa r à l'additionneur 26.

25 L'additionneur 26 additionne ainsi séquentiellement bit par bit l'aléa r et le résultat fourni par le circuit de multiplication 22.

Les différents moyens inclus dans le dispositif 10 sont réalisés en logique câblée.

Un tel dispositif 10 est monté, tel que décrit à la figure 8, sur un support 28 au format d'une carte de crédit par exemple. Le support 28 peut par 5 exemple être inséré dans un lecteur 30 hébergeant l'application de sécurité 34.

Dans un exemple d'application, l'insertion du support 28 dans le lecteur 30 active automatiquement l'application de sécurité 34 qui sollicite la puce électronique et lui transmet des données, particulièrement le challenge c . La puce électronique s'authentifie en lui fournissant une valeur cryptographique y 10 (ou V) calculée selon le procédé de l'invention. Comme décrit à la figure 1, un paramètre x lié à r est transmis à l'application de sécurité.

A partir des valeurs x et V (ou y), l'application de sécurité procède alors à la vérification.

De façon classique, lorsque le paramètre x est de la forme $x = g^r$, 15 l'équation vérifiée par l'application de sécurité peut être de la forme $g^y = x \cdot p^c$ lorsque la valeur cryptographique est de la forme $y = r + sc$, ou $g^y \cdot p^c = x$ si la valeur cryptographique est de la forme $y = r - sc$.

Dans le cas où le paramètre x fait apparaître une fonction de hachage $(x = h(g^r, D))$, l'équation de vérification est classiquement $h(g^y \cdot p^c, D) = x$ lorsque 20 la valeur cryptographique est de la forme $y = r + sc$. De préférence, on utilise alors une valeur cryptographique de la forme $y = r - sc$, afin que l'équation de vérification ne fasse pas intervenir de divisions : $h(g^y \cdot p^c, D) = x$.

Lorsque la valeur d'authentification est validée, l'authentification de la 25 puce est correcte et l'application de sécurité en informe la puce électronique. Les transactions entre l'application de sécurité et la puce électronique sont ainsi autorisées.

REVENDEICATIONS

1. Procédé pour accomplir une opération cryptographique dans un dispositif (10) sous le contrôle d'une application de sécurité (34), dans lequel on produit une valeur cryptographique (y) dans le dispositif, par un calcul comprenant au moins une multiplication entre deux facteurs incluant une partie au moins d'une clé secrète (s) associée au dispositif, caractérisé en ce que, un premier des deux facteurs de la multiplication ayant un nombre de bits déterminé L en représentation binaire, on constraint le second des deux facteurs de la multiplication pour qu'il comprenne, en représentation binaire, plusieurs bits à 1 avec, entre chaque paire de bits à 1 consécutifs, une séquence d'au moins L-1 bits à 0, et en ce que la multiplication est réalisée en assemblant des versions binaires du premier facteur respectivement décalées conformément aux positions des bits à 1 du second facteur.
2. Procédé selon la revendication 1, dans lequel la clé secrète (s) fait partie d'une paire de clés cryptographiques asymétriques associée au dispositif (10).
3. Procédé selon la revendication 1 ou 2, dans lequel le dispositif (10) comprend une puce incluant une logique câblée pour produire la valeur cryptographique.
4. Procédé selon l'une quelconque des revendications précédentes, dans lequel le calcul de la valeur cryptographique comprend en outre une addition ou une soustraction entre un nombre pseudo-aléatoire (r) et le résultat de la multiplication.
5. Procédé selon la revendication 4, dans lequel les premier et second facteurs (s, c) et le nombre pseudo-aléatoire (r) sont dimensionnés pour que le nombre pseudo-aléatoire soit supérieur au résultat de la multiplication.

6. Procédé selon la revendication 5, dans lequel le nombre de bits à 1 du second facteur est choisi au plus égal au plus grand entier inférieur ou égal à s_1/L , où s_1 est un seuil prédéfini inférieur au nombre de bits du nombre pseudo-aléatoire (r) en représentation binaire.

5 7. Procédé selon l'une quelconque des revendications précédentes, dans lequel les deux facteurs de la multiplication incluent, en plus de ladite partie de la clé secrète (s), un nombre (c) fourni au dispositif par l'application de sécurité exécutée en dehors du dispositif.

8. Procédé selon l'une quelconque des revendications 1 à 6, dans 10 lequel les deux facteurs de la multiplication incluent, en plus de ladite clé secrète (s), un nombre (c) fourni par le dispositif.

9. Procédé selon l'une quelconque des revendications précédentes, dans lequel ladite partie de la clé secrète (s) est ledit premier facteur de la multiplication.

15 10. Procédé selon la revendication 9, dans lequel lesdites versions binaires sont disposées dans des intervalles respectifs de même taille en bits, ladite taille correspondant à la taille totale d'un espace utilisable divisée par le nombre de bits à 1 du second facteur de la multiplication, chaque version binaire étant placée dans son intervalle respectif en fonction d'un décalage 20 conformément aux positions des bits à 1 du second facteur.

11. Procédé selon l'une quelconque des revendications 1 à 8, dans lequel ladite partie de la clé secrète (s) est le second facteur de la multiplication.

25 12. Procédé selon la revendication 11, dans lequel la clé secrète (s) est stockée dans un support de mémoire du dispositif en codant les positions de ses bits à 1.

13. Procédé selon l'une quelconque des revendications 11 à 12, dans lequel la clé secrète (s) est stockée dans un support de mémoire (16) du dispositif en codant des nombres de bits séparant respectivement des bornes inférieures d'intervalles de $(S-1)/(n-1)$ bits et des bornes inférieures de blocs 5 de bits alloués au premier facteur (c) de la multiplication et disposés chacun dans les intervalles associés, S étant le nombre de bits de la clé secrète et n le nombre de bits à 1 de la clé secrète.

14. Procédé selon l'une quelconque des revendications 11 à 12, dans lequel la clé secrète (s) est stockée dans un support de mémoire (16) du dispositif en codant des nombres de bits, chacun représentatif du nombre de 10 bits séparant deux blocs de bits successifs alloués au premier facteur (c) de la multiplication.

15. Procédé selon l'une quelconque des revendications précédentes, dans lequel la valeur cryptographique (y) est produite pour authentifier le dispositif dans une transaction avec l'application de sécurité exécutée en 15 dehors du dispositif.

16. Procédé selon l'une quelconque des revendications 1 à 14, dans lequel la valeur cryptographique (y) est produite en tant que signature électronique.

20 17. Dispositif à fonction cryptographique, comprenant des moyens (24) d'interface avec une application de sécurité (34) et des moyens de calcul (12, 22, 26) pour produire une valeur cryptographique (y), les moyens de calcul comprenant des moyens de multiplication (22) entre deux facteurs incluant une partie au moins d'une clé secrète (s) associée au dispositif, caractérisé en ce 25 que, un premier des deux facteurs de la multiplication ayant un nombre de bits déterminé L en représentation binaire, et le second des deux facteurs de la multiplication étant contraint pour comprendre, en représentation binaire, plusieurs bits à 1 avec, entre chaque paire de bits à 1 consécutifs, une séquence d'au moins L-1 bits à 0, les moyens de multiplication comprennent

des moyens pour assembler des versions binaires du premier facteur respectivement décalées conformément aux positions des bits à 1 du second facteur.

18. Dispositif selon la revendication 17, comprenant en outre des 5 moyens (12) de génération d'un nombre pseudo-aléatoire (r), les moyens de calcul comprenant des moyens (26) pour ajouter ou soustraire le résultat de la multiplication audit nombre pseudo-aléatoire.

19. Dispositif selon la revendication 18, dans lequel les premier et second facteurs (s, c) et le nombre pseudo-aléatoire (r) sont dimensionnés 10 pour que le nombre pseudo-aléatoire soit supérieur au résultat de la multiplication.

20. Dispositif selon l'une quelconque des revendications 17 à 19, dans lequel les moyens (12,22,26) de calcul sont réalisés en logique câblée.

21. Dispositif selon l'une quelconque des revendications 17 à 20, dans 15 lequel ladite partie de la clé secrète (s) est le premier facteur de la multiplication.

22. Dispositif selon l'une quelconque des revendications 17 à 20, dans lequel ladite partie de la clé secrète (s) est le second facteur de la multiplication.

20 23. Dispositif selon la revendication 22, comprenant en outre une mémoire (16) adaptée pour stocker des données de codage des positions des bits à 1 de la clé secrète (s).

FIG. 1

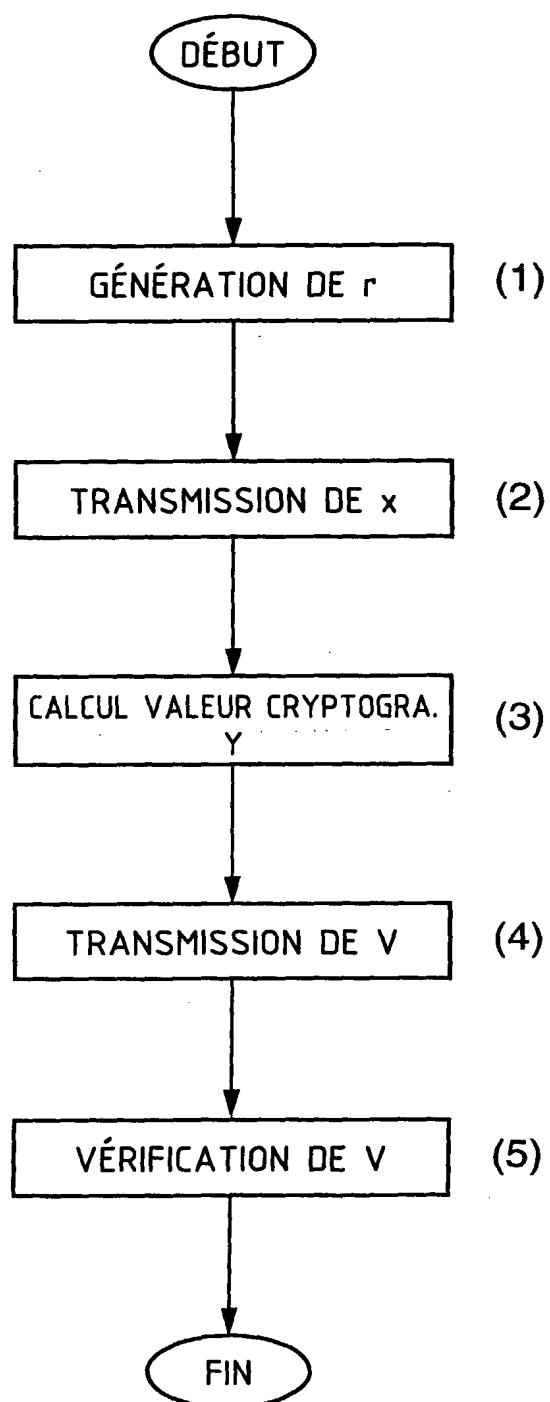


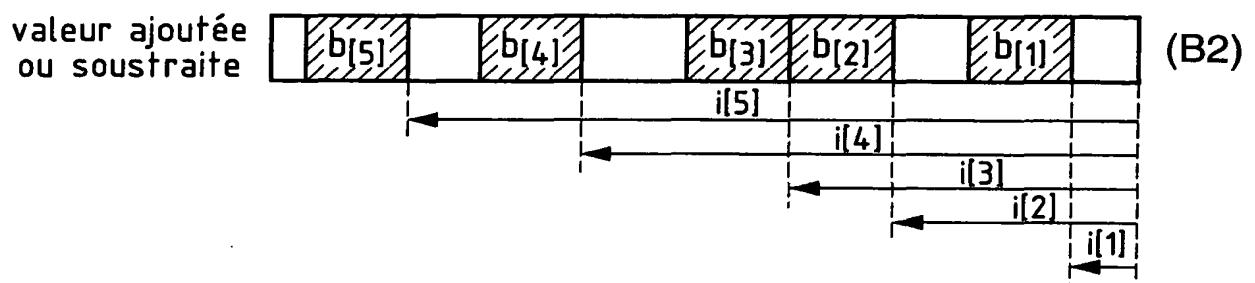
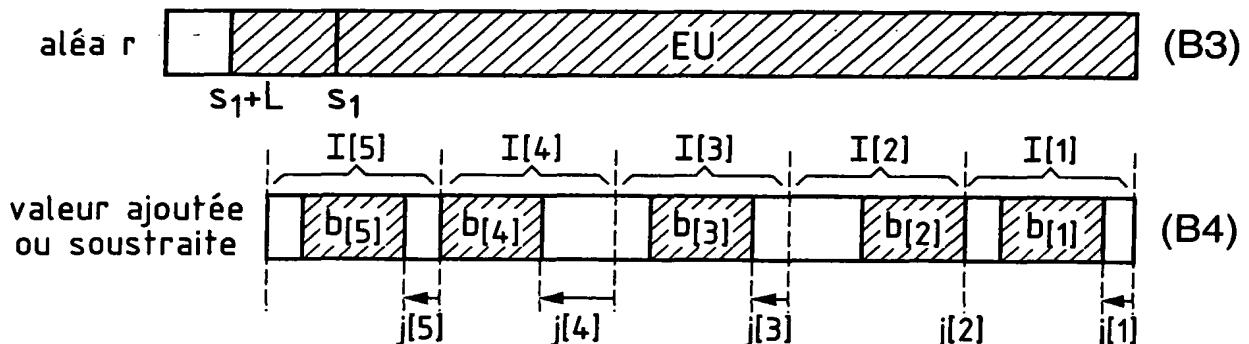
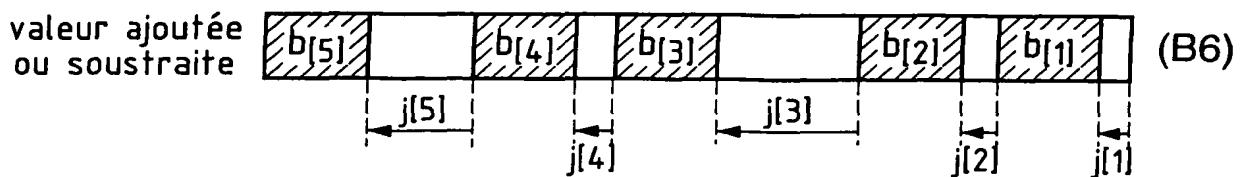
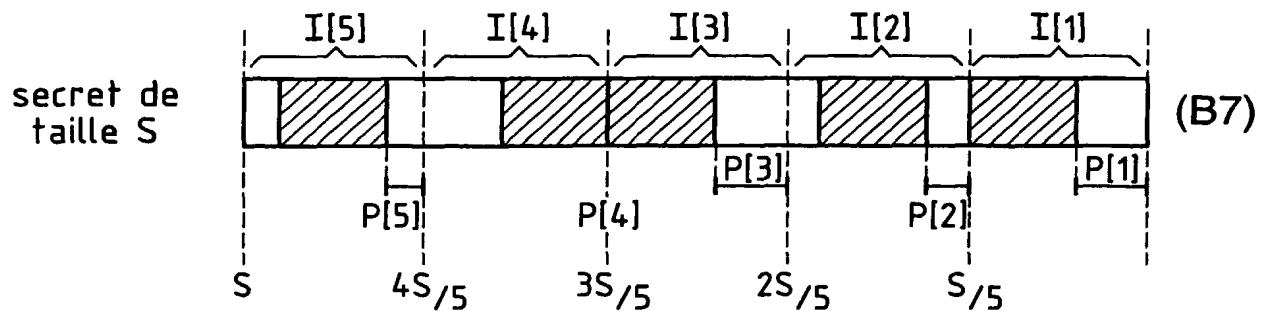
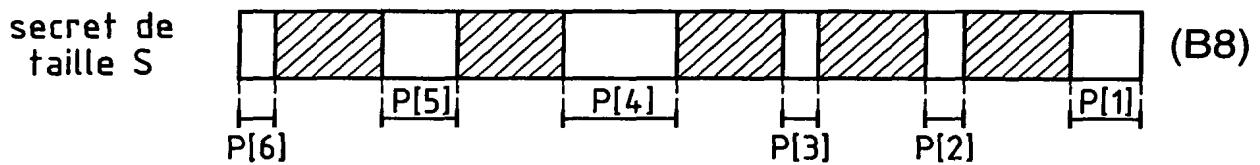
FIG. 2**FIG. 3****FIG. 4**

FIG. 5**FIG. 6**

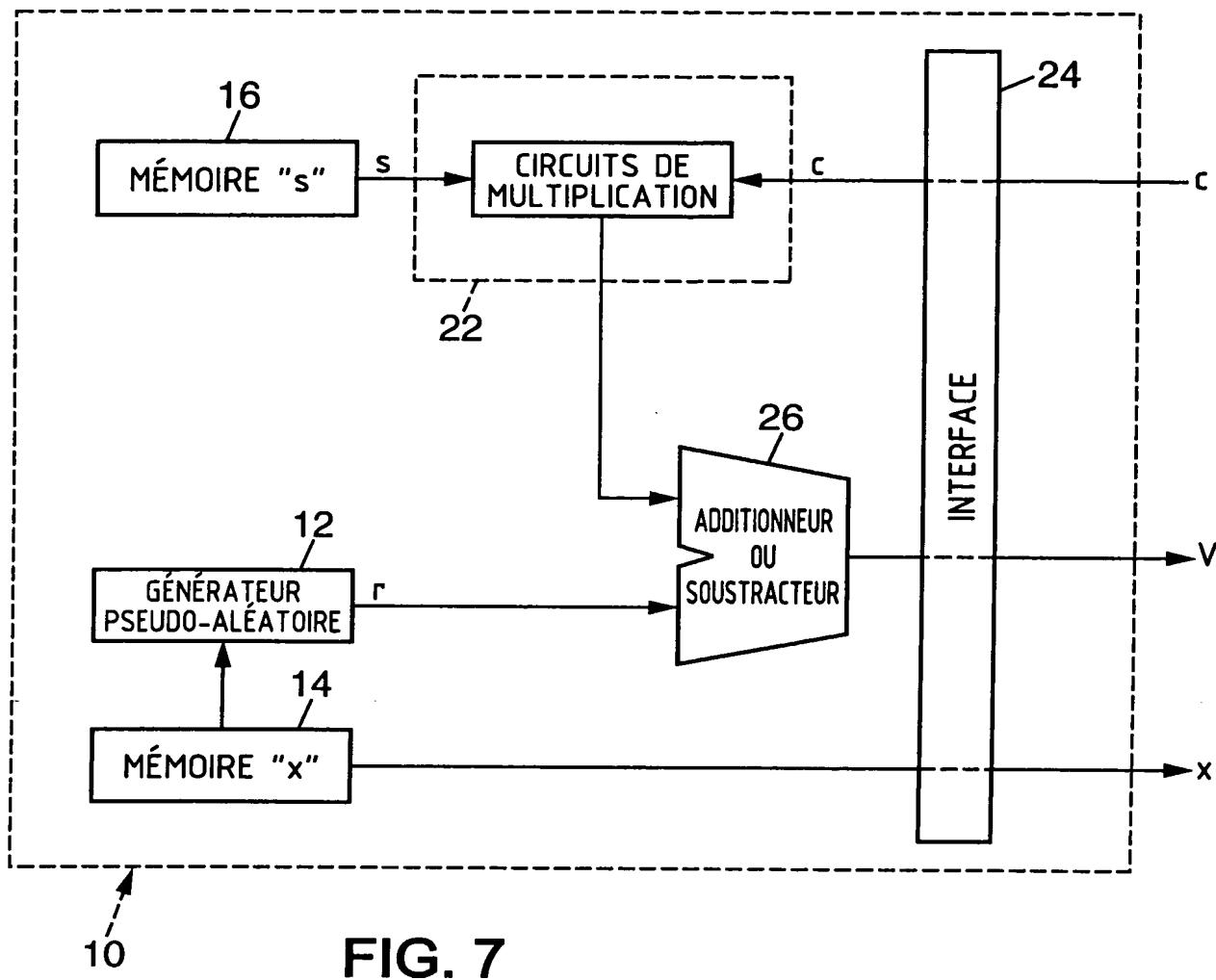


FIG. 7

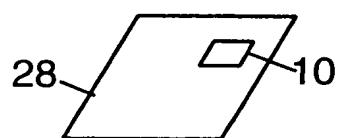
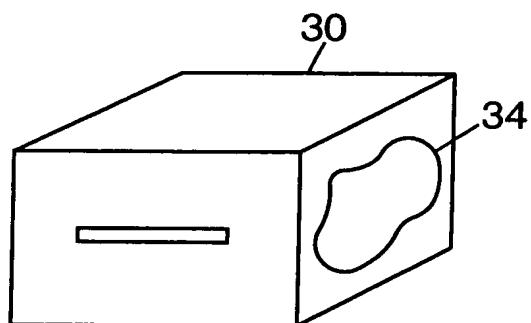


FIG. 8